

APPLICATION  
FOR  
UNITED STATES LETTERS PATENT

TITLE: EXPEDITED DYNAMIC MIRROR SERVICE POLICY  
APPLICANT: YECHIEL YOCHAI, RON ARNAN AND SACHIN MORE

CERTIFICATE OF MAILING BY EXPRESS MAIL

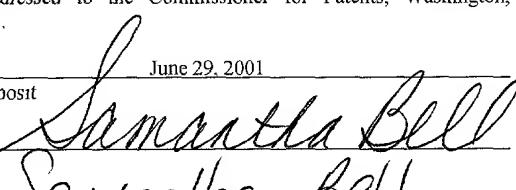
Express Mail Label No. EL445350002US

I hereby certify under 37 CFR §1.10 that this correspondence is being deposited with the United States Postal Service as Express Mail Post Office to Addressee with sufficient postage on the date indicated below and is addressed to the Commissioner for Patents, Washington, D C. 20231.

June 29, 2001

Date of Deposit

Signature



Typed or Printed Name of Person Signing Certificate

EXPEDITED DYNAMIC MIRROR SERVICE POLICY

5

BACKGROUND OF THE INVENTION

The invention relates generally to data storage systems, and in particular, to data storage systems in which 10 data is stored on mirrored logical volumes.

Various techniques have been developed for increasing the reliability of data storage systems and to protect against failure due to data loss. One such technique is data mirroring, which involves the storage of two or more copies of the data, preferably on different storage devices. A data storage having mirrored data operates to maintain all copies up to date. Typically, there are two copies.

In a typical data storage system, each physical device, such as a disk storage device, may have a format that defines a number of units, and the units may correspond to 20 logical volumes. Each logical volume appears to the host computer or computers to be a separate storage device and, therefore, operations take place with regard to the logical volume as if it were a physical volume. Typically, the 25 physical devices are configured as mirrored pairs, with each physical device in the pair storing corresponding copies of logical volumes. Such a data storage system is said to employ both mirrored physical devices and mirrored logical copies.

Some logical volumes are accessed for read and/or 30 write operations substantially more often than other logical volumes, and the nature of the operations can vary from random

accesses to sequential accesses during any particular sequence of read or write operations. As a result, with larger storage devices, the risk of having concurrent jobs, each pertaining to a relatively small file (local activity) interfering with each other's activity increases. The use of striping techniques further increases the risk. Depending upon the nature and frequency of the read and write operations to a storage device containing many logical volumes, the disk read/write heads may be severely worked and response time can, on average, be increased if sequential read and write locations are not located near one another.

One prior solution sets a mirror service policy, statically or preferably dynamically (as described in U.S. Patent No. 6,112,257, the contents of which are incorporated herein, by reference), in an attempt to optimize the reading and writing of data. Such a policy may, for example, assume that most data transfers are sequential in nature, or that they are random.

One such policy is to only service requests that fall within the first half of a logical volume on one of a mirrored pair of physical devices and the requests to the second half of the logical volume on the other, mirrored physical device. Another policy might be to only service requests that fall in odd groups with a group being defined in terms of a number of blocks. For instance the first group (blocks 0-959) will be serviced by copy 1 and the next group by copy 2, etc.

The approach described in the above-referenced patent

sorts logical volumes by activity level, load balances the physical devices based on the activity levels of the logical volumes and minimizes seeks for the logical volumes on the same physical devices.

5 Because volumes do not necessarily reside on mirrored physical devices, it is desirable, also, to ensure load balancing over the non-mirrored physical devices. Another technique, described in U.S. application Serial No. 09/382,752, filed August 25, 1999, and entitled Dynamic Mirror 10 Service Policy With Seek Adjustment in a Non-Physical Mirrored Storage Environment, the contents of which are incorporated herein, by reference, provides for a seek minimization to adjust seek activity when logical volumes, but not necessarily physical devices, are mirrored.

## SUMMARY OF THE INVENTION

In one aspect of the invention, response time of physical devices in a data storage system is optimized by collecting statistics for each of the physical devices, determining from the statistics the  $n$  most active of the physical devices and, for each of the  $n$  most active of the physical devices, adjusting a mirror service policy associated with one or more mirrored logical volumes serviced by the physical device to reduce seek time.

Embodiments of the invention may include one or more of the following features.

The physical device statistics include utilizations and adjusting is performed if the utilization of the physical device is greater than a threshold value.

One or more aspects of the invention may include one or more of the following advantages. Activity level sorts are performed on physical devices (and only the n busiest of the physical devices), instead of the logical volumes in a data storage system. Thus, the time-consuming logical volume based sorting and load balancing of the physical devices on a logical volume by logical volume basis (as performed by prior approaches) is eliminated. Consequently, seek (and, therefore, response time) optimization based on dynamic mirror service policy adjustment can be performed on a more frequent basis and so adjusts more quickly to changes in workload.

Other features and advantages of the invention will be apparent from the following detailed description and from the claims.

#### BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram of a data storage system in which the invention can be employed.

FIG. 2 is a detailed block diagram of the storage controller shown in FIG. 1.

FIG. 3 is a block diagram of the disk director shown in FIG. 2.

FIGS. 4 is flow diagrams illustrating an expedited

mirror service policy process for the disk director shown in FIG. 3.

FIG. 5 is an example of simulated policy changes as performed by the process (of FIG. 4) for physical mirrors.

5 FIG. 6 is an example of simulated policy changes as performed by the process (of FIG. 4) for non-physical mirrors.

FIG. 7 is a graphical depiction of response time versus time for the expedited DMSP process in comparison to a prior DMSP processing scheme.

10  
15  
20  
DETAILED DESCRIPTION

Referring to FIG. 1, a data processing system 10 includes host computers 12a, 12b, ..., 12m, connected to a data storage system 14. The data storage system 14 receives data and commands from, and delivers data and responses to, the host computers 12. The data storage system 14 is a mass storage system having a controller 16 coupled to pluralities of physical storage devices shown as disk devices 18a, disk devices 18b, ..., disk devices 18k. Each of the disk devices 18 is logically divided, in accordance with known techniques, into one or more logical volumes.

The controller 16 interconnects the host computers 12 and the disk devices 18. The controller can be, for example, a controller of the Symmetrix data storage system from EMC Corporation. Although described herein as a component of a data storage system, the controller 16 could also be a

separate appliance or server. The controller 16 thus receives  
memory write commands from the various host computers over  
buses 20a, 20b,..., 20m, respectively, for example, connected  
and operated in accordance with a SCSI protocol, and delivers  
5 the data associated with those commands to the appropriate  
devices 18a, 18b,..., 18k, over respective connecting buses  
22a, 22b,..., 22k. Buses 22 also operate in accordance with a  
SCSI protocol. Other protocols, for example, Fibre Channel,  
could also be used for buses 20, 22. The controller 16 also  
receives read requests from the host computers 12 over buses  
10 20, and delivers requested data to the host computers 12,  
either from a cache memory of the controller 16 or, if the  
data is not available in cache memory, from the disk devices  
18.

15 In a typical configuration, the controller 16 also  
connects to a console PC 24 through a connecting bus 26. The  
console PC 24 is used for maintenance and access to the  
controller 16 and can be employed to set parameters of the  
controller 16 as is well known in the art.

20 Optionally, the controller may be connected to a  
remote data processing system like the data processing system  
10 or a remote data storage system like the data storage  
system 14 (shown in dashed lines) for data back-up capability  
by a data link 28. The data link may be implemented as an  
25 ESCON fiber-optic link, or may be connected and operated in  
accordance with Fibre Channel or Gigabit Ethernet protocols.  
Other types of remote connections can also be used. The data

link 28 enables a remote data storage system to store on its own devices a copy of information stored in the devices 18 of the data storage system 14 in a mirrored manner.

In operation, the host computers 12a, 12b, ..., 12m,  
5 send, as required by the applications they are running,  
commands to the data storage system 14 requesting data stored  
in the logical volumes or providing data to be written to the  
logical volumes. Referring to FIG. 2, and using the EMC  
Symmetrix controller as an illustrative example, details of  
10 the internal architecture of the data storage system 14 are  
shown. The communications from the host computer 12 typically  
connect the host computer 12 to a port of one or more host  
directors 30 over the SCSI bus lines 20. Each host director,  
in turn, connects over one or more system buses 32 or 34 to a  
15 global memory 36. The global memory 36 is preferably a large  
memory through which the host director 30 can communicate with  
the disk devices 18. The global memory includes a common area  
38 for supporting communications between the host computers 12  
and the disk devices 18, a cache memory 40 for storing data  
20 and control data structures, and tables 42 for mapping areas  
of the disk devices 18 to areas in the cache memory 40.

Also connected to the global memory 36 are back-end  
(or disk) directors 44, which control the disk devices 18. In  
the preferred embodiment, the disk directors are installed in  
25 the controller 16 in pairs. For simplification, only two disk  
directors, indicated as disk directors 44a and 44b, are shown.  
However, it will be understood that additional disk directors

may be employed by the system.

Each of the disk directors 44a, 44b supports four bus ports. The disk director 44a connects to two primary buses 22a and 22b, as well as two secondary buses 22a' and 22b'.

- 5 The buses are implemented as 16-bit wide SCSI buses. As indicated earlier, other bus protocols besides the SCSI protocol may be used. The two secondary buses 22a' and 22b' are added for redundancy. Connected to the primary buses 22a, 22b, are the plurality of disk devices (e.g., disk drive units) 18a and 18b, respectively. The disk director 44b connects to two primary buses 22c and 22d. Connected to the primary buses 22c, 22d are the plurality of disk devices or disk drive units 18c and 18d. Also connected to the primary buses 22c and 22d are the secondary buses 22a' and 22b'. When the primary bus is active, its corresponding secondary bus is inactive, and vice versa. The secondary buses of the disk director 44b have been omitted from the figure for purposes of clarity.

- Like the host directors 30, the disk directors 44 are also connected to the global memory 36 via one of the system buses 32, 34. During a write operation, the disk directors 44 read data stored in the global memory 36 by a host director 30 and write that data to the logical volumes for which they are responsible. During a read operation and in response to a read command, the disk directors 44 read data from a logical volume and write that data to global memory for later delivery by the host director to the requesting host computer 12.

As earlier mentioned, the data storage system 14 may be remotely coupled to another data storage system 14 in a mirrored storage configuration via the data link 28. Still referring to FIG. 2, each data storage system 14 in the 5 mirrored storage configuration includes a remote director 48 to connect to the data link 28 and handle transfers of data over that link. The remote director 48 communicates with the global memory 36 over one of the system buses 32, 34.

As is known in the art, mirrors of logical volumes may 10 be maintained in conjunction with mirrors of physical devices.

The logical volume mirrors are copies of each other and the physical device mirrors are copies of each other. For example, if there are two mirrored physical devices M1 and M2, four mirrored logical volumes A, B, C, D and M1 stores logical 15 volumes A through D only, then M2 stores logical volumes A through D only. It is also possible to support logical volume mirrors on non-mirrored physical devices. In such a non-physical mirrored configuration, all of the corresponding mirrored copies of the logical volumes stored on one of the 20 non-mirrored physical devices will not reside together on a different physical device. Examples of a physical mirrored arrangement and a "non-physical" mirrored arrangement (i.e., logical volume mirrors stored on non-mirrored physical devices) are later shown in FIGS. 5 and 6, respectively.

With reference to the data storage system 14, it is 25 assumed that mirroring exists at the logical volume level, but not necessarily at the physical device level. Typically, for

purposes of this embodiment, paired logical volume copies are employed and it is assumed that each unit stores a logical volume. The controller 16 can then respond to a read request by reading from either of the logical volume copies or mirrors stored in the physical devices 18. Accordingly, there is implemented in the controller 16, and among the disk adapters 40, a mirror service policy which identifies which of the device adapters 40 is primarily responsible for reading in response to requests from a host computer. This policy could be fixed at the beginning of a system set up, taking into account the expected loads. Preferably, however, the mirror service policy implementation is dynamic. That is, the mirror service policy can be modified, periodically, in response to statistics describing the nature of the read and write requests to the data storage system 14, to change the mirror service policy during the operation of the data storage system 14.

The statistics are collected over a given period. The information collected includes access rates for both read and write operations and how sequential the accesses were during the time period. Many factors are included in the policy decisions. These factors include: the activity level of the physical devices involved, the activity level of the disk directors involved, seek and transfer times, physical device utilization (based on the potential and actual transfer rates of a physical device), and how sequential the accesses were.

As shown in FIG. 3, the directors 30, 44 and 48

(represented in the figure by the director 44) include a processor 50 coupled to a control store 51 and a local, nonvolatile memory (NVM) 52 by an internal bus 54. The processor 50 controls the overall operations of the director 44 and communications with the memories 51 and 52. The local memory 52 stores firmware (or microcode) 56, data structures 58, as well as parameter/variable data in a parameter store 60. The disk director 44 may include more than one processor 50 for parallel or distributed processing within a particular disk director.

Although not shown, it will be understood by those skilled in the art that the disk director 44 will include I/O interface logic, as well as bus and control logic. In addition, the disk director 40 may include transmit/receive buffers for temporarily storing data to be transferred to and received from the global memory 36.

The firmware 56, data structures 58 and parameter store 60 are read each time the data storage system 14 is initialized. The microcode 56 is copied into the control store 51 at initialization for subsequent execution by the processor 50.

While the director firmware 56, data structures 58 and parameter data of the parameter store 60 are preferably resident on the controller 16 (as described above), they may 25 be partly resident on the PC 24 or the host 12.

The components of the director microcode 56 include the following: a system calls/host application layer 62;

advanced functionality modules 64, which may be optional at the director level or even at the data storage subsystem level; common function modules 66, which are provided to each director; an interface module 68; and one or more physical transport (or device) drivers 70. The common function modules 66 includes a number of processes executed by the processor 50 to control data transfer between the host computer 12, the global memory 36 and the disk devices 18, e.g., a cache manager having routines for accessing the cache memory 40 and associated tables 42.

Interface modules exist for each of the different types of directors that are available based on connectivity and/or function and thus define the director functionality. Specifically, for the disk director 44, the interface module 68 is a disk interface module. That is, a director that has been loaded with the disk interface code 68 is thus programmed to serve as the disk director 44 or one of disk directors 44 (when more than one is present in the system). As such, it is responsible for controlling back-end operations of the controller 16. The disk interface module 68 includes an expedited Dynamic Mirror Service Policy (DMSP) process 72, as will be described. Also supported by the disk interface module 68 are services for read misses, write destaging, read prefetching, RAID, data copy, as well as other background drive operations (not shown).

The data structures 58 include DMSP tables for storing statistics, dynamic mirror service policies for each pair of

logical volume mirrors and DMSP processing computation results.

The process 72 employs a cost function to estimate service time for each physical device in the data storage system. More specifically, the cost function for a physical device is defined as the seek time (a function of logical volume distribution) plus transfer time (a function of size of I/Os) for that physical device.

Referring to FIG. 4, a flow diagram illustrating the expedited DMSP process 72 is shown. The process 72 begins 80 by collecting statistics such as activity level and utilization for all of the physical devices (step 82). Once the time has come to reconsider logical volume mirror policies, the statistics are reviewed to determine the top  $n$  most active or busiest of the physical devices (step 84). The  $n$  busiest physical devices are sorted by activity level from most active to least active (step 86). The process 72 selects the first physical device on the sorted list, that is, the busiest disk (step 88). The process 72 temporarily assigns volumes in the first half of the selected physical device to itself and the remainder to corresponding mirror or mirrors (step 90). This change serves to minimize seeks, but does not guarantee load balance. The process computes or determines the utilization of the selected physical device with this temporary DMSP policy (step 92). The process 72 determines if the utilization of the selected physical device is less than a utilization threshold (step 93). In the described embodiment,

the utilization threshold is 50%. The value of n and the utilization threshold are user-defined and are included in the DMSP parameter data 74 stored in the parameter store 60 (from FIG. 3).

5 If it is determined that the utilization is not greater than the threshold, in this case, 50% utilization, the process 72 makes the temporary policy permanent (step 94) and determines whether or not there is another physical device on the sorted list (step 96). If the process 72 determines that 10 it has come to the end of the sorted list, it terminates (step 97). If there is at least one more physical device on the sorted list, the process 72 selects the next (busiest) physical device on the list (step 98).

Otherwise, that is, if it is determined that the utilization is greater than the threshold, the process 72, based on the temporary policy as described above, selects a first, outermost logical volume serviced by the selected physical device (the logical volume adjacent to a border between the logical volume(s) serviced by the selected 15 physical device and logical volume(s) serviced by another physical device) and its corresponding mirrored copy (step 100). The process 72 thus determines which one or more of the other physical devices store a mirrored copy corresponding to one of the logical volumes (including the selected logical 20 volume) serviced by the selected physical device. As will be discussed in further detail later, if the selected physical device is physically mirrored, then there will only be one 25

other physical device involved in any DMSP  
adjustment/evaluation for the selected physical device. If  
the mirrored copies of the logical volumes handled by the  
selected physical device reside on more than one physical  
5 device, then the DMSP update of the selected physical device  
will involve more than one other physical device. The process  
72 computes the cost functions for the selected physical  
device as well as the physical device(s) having mirrored  
copies corresponding to the logical volumes currently assigned  
10 to the selected physical device (step 102) and determines the  
maximum of the computed cost functions (step 104). The  
process 72 simulates a first policy change for the selected  
physical device by reassigning the preferred device mirror  
policy from the selected logical volume to the corresponding  
15 mirrored copy (step 106). At this stage, and as indicated  
above, the process is merely simulating the change, that is,  
the reassignment has not actually been implemented by  
recording the policy "swap" in the DMSP tables 82 (as described  
in the above-referenced U.S. applications). Rather, the  
20 reassignment information, as well as the various computation  
results, is stored in processor memory. The process 72 then  
recomputes the cost function values for the physical devices  
involved in the first simulated policy change (step 108) and  
determines a new maximum cost function as the maximum of the  
25 recomputed values under the first policy evaluation (step  
110).

The process 72 then considers a second policy change

that simulates an interleaved move. That is, the process 72 simulates moving a second half of the selected logical volume (that is, the half nearest the border) to the other physical device (step 112). The process 72 recomputes the cost  
5 functions (step 114) and determines a new maximum cost function value for the 2<sup>nd</sup> policy change (step 116).

At this point, the process 72 has generated three maximum cost function values, one value for no change in policy (that is, using the temporary policy of step 90), a  
10 second value for the first policy change (change mirror for volume) and a third number for the second policy change (interleaved move). The policy chosen for the logical volume under consideration is the minimum of these three values.  
Thus, the process 72 determines the minimum of the three  
15 maximum cost function values (step 118). The process records the policy change corresponding to the determined minimum value in the DMSP tables (step 120). Because the process 72,  
is starting off with the temporary policy when it evaluates  
the first and second policy evaluations on the temporary  
20 policy, the policy that is recorded is the temporary policy as modified by the policy change corresponding to the minimum unless that policy change is the temporary policy itself, in which case the process 72 records the temporary policy as the new policy.

25 If the minimum corresponds to the first policy change (step 122), the process 72, proceeding to move the border further inward, i.e., in a sequential manner from the

outermost to innermost of logical volumes serviced by the selected physical device, determines if there is another logical volume (serviced by the physical device) on the physical device (step 124). If so, the process 72 proceeds to 5 the next logical volume in the sequence (step 126) and returns to step 102.

If the logical volume is the last of any logical volumes serviced by the physical device, the process 72 returns to step 96 to determine if another of the n busiest physical devices on the sorted listed is to be evaluated. 10

If the minimum of the maximum cost function values corresponds to either the interleaved policy change or no change in policy, the process 72 returns to the step of determining if a next busiest disk on the sorted list is to be evaluated (step 96). 15

FIG. 5 provides an example of the iterative simulated policy changes involving two mirrored ones of the physical devices 18, shown as 18a-1 ("M1") and 18a-2 ("M2"). Referring to FIG. 5, the physical devices M1 and M2 store a plurality of 20 logical volumes 130. In the example shown, the physical devices M1 and M2 are configured to store eight logical volumes apiece, for a total of eight logical volumes mirrored pairs. Under the temporary policy (of step 90 in FIG. 4), the first half of the logical volumes (LV<sub>1</sub> to LV<sub>4</sub>) are serviced by 25 M1 and the second half of the logical volumes (LV<sub>5</sub> to LV<sub>8</sub>) are serviced by M2. Thus, the border (indicated by the solid arrow) for the logical volumes serviced by M1 is determined to

be between LV<sub>4</sub> and LV<sub>5</sub>. It is assumed for this example that the percentage of sequential I/Os seen by M1 is 80 and the percentage of sequential I/Os seen by M2 is 20.

Referring to FIGS. 4 and 5, the process 72 selects LV<sub>4</sub> 5 to be moved from M1 to M2 (step 100). It computes the respective initial cost function values C(M1) and C(M2) of the physical devices M1, M2 as 100 and 50, respectively (step 102), and determines a maximum of the computed values, that is, 100 (step 104). After the simulated first policy move 10 (step 106), the process 72 recomputes the cost functions C(M1) and C(M2) as 80 and 70, respectively (step 108), and determines 80 to be the maximum value (step 110). The process 72 also evaluates an interleaved move (step 112), that is, moving only the second half of LV<sub>4</sub> from M1 to M2. The process 15 72 computes C(M1) and C(M2) for the interleave as 90 and 75, respectively (step 114), with a maximum value of 90. As the maximum value of 80 for the first policy change is the minimum of the three maximum cost function values, the process 72 proceeds to evaluate LV<sub>3</sub> for transfer from M1 to M2 (step 20 126).

In this second pass, after moving LV<sub>3</sub> from M1 to M2, the cost functions C(M1) and C(M2) are computed to be 60 and 80, respectively, and the new maximum is determined to be 90. The process 72 evaluates an interleaved move, that is, moving 25 only the second half of LV<sub>3</sub> from M1 to M2. The process 72 computes C(M1) and C(M2) for the interleave as 70 and 75, respectively. As the new maximum of 75 is less than the

maximums of 85 (first policy change) and 80 (no change), the process 72 records the policy changes for LV<sub>4</sub> and LV<sub>3</sub> (interleave) so that LV<sub>4</sub> and the second half of LV<sub>3</sub> will be serviced by M2 instead of M1 (step 126).

5 As indicated earlier, the physical devices need not be mirrored. FIG. 6 illustrates a non-physical mirrored arrangement. In the example shown, there are four non-mirrored physical devices 18b-1 ("P1"), 18b-2 ("P2"), 18b-3 ("P3"), 18b-4 ("P4"). As in the example shown in FIG. 5, the 10 selected one of the n busiest disks, in this case, "P1", services logical volumes LV<sub>1</sub> through LV<sub>3</sub>. The logical volumes LV<sub>4</sub> through LV<sub>6</sub> are handled by P4. The logical mirrors for LV<sub>1</sub> through LV<sub>3</sub> reside on P2 and P3. More specifically, the logical mirrored pair for LV<sub>3</sub> resides on P1 and P3, and the logical mirrored pairs for LV<sub>2</sub> and LV<sub>1</sub> involve P1 and P2.

Referring to FIGS 4 and 6, the processing for the non-physical mirrored arrangement is much the same as it is for a physical mirrored arrangement (such as the arrangement depicted in FIG. 5) except that maximum cost function 20 determinations must take into account all of the physical devices on which logical mirrored pairs reside. For the example shown in FIG. 6, the physical device P1 is the physical device under evaluation. It includes six logical volumes, LV<sub>1</sub> through LV<sub>6</sub>. According to the existing mirror service policy, LV<sub>1</sub> through LV<sub>3</sub> are serviced by P1 and LV<sub>4</sub> through LV<sub>6</sub> are serviced by P4. Mirrored copies of LV<sub>1</sub> and LV<sub>2</sub> are maintained by P2 and a mirrored copy of LV<sub>3</sub> is maintained

by P3. If the process 72 determines that the utilization of P1 is less than the threshold, then the border between LV<sub>3</sub> and LV<sub>4</sub> remains fixed, that is, P1 will service LV<sub>1</sub> through LV<sub>3</sub>, and P4 will service LV<sub>4</sub> through LV<sub>6</sub> (at least until the next 5 updating occurs). If the utilization is determined to be greater than the threshold, the process attempts to move the border in the direction of the solid arrow, one logical volume (or half volume, in the interleaved case) at a time). Thus, the process 72 tries to reassign the work for LV<sub>3</sub> to P3, and 10 possibly the work for LV<sub>1</sub>/LV<sub>2</sub> to P2, in the manner described above, with the objective of reducing the worst case cost function with each successive iteration until the worst case cost function can be reduced no further.

For each iteration, the maximum cost function value is 15 taken of the physical devices affected. For example, if LV3 is being adjusted, then the maximum value is determined between the values for the cost function C(P1) and the cost function C(P3). As was described above with reference to FIG. 4, the process 72 computes a maximum for each of the three 20 policies and adopts the policy that provided the value determined to be the minimum value.

It should be noted that once the border for a particular physical device has been moved in one direction as a result of the expedited DMSP processing for that physical 25 device, its location is fixed for the remainder of the processing. The border cannot then be moved in the other direction when logical volumes residing on that physical

device are involved in a subsequent evaluation of another one  
of the  $n$  busiest physical devices. Thus, the process 72  
recognizes that a physical device involved in an evaluation of  
a different physical device has been evaluated itself already  
5 and makes no attempt to transfer volume activity to that  
physical device.

The illustration of FIG. 7 depicts the responsiveness  
of the expedited DMSP process (curve indicated by reference  
numeral 140) relative to a prior DMSP process that  
10 sorts/processes by logical volume and for all logical volumes  
in all physical devices in the system (curve indicated by  
reference numeral 142) in terms of response time ("RT") as a  
function of time. Because the old scheme has to perform a  
sort for and evaluate all physical devices logical volume by  
15 logical volume, it can only make response time improvements  
(based on DMSP statistics-based processing updates) at  
intervals of time "t". The value of  $t$  can be as great as 5  
minutes. In contrast, the expedited DMSP process 72, which  
evaluates only the  $n$  busiest of the physical devices and  
20 therefore performs policy updates at a much greater frequency  
(for example, every 20 seconds), adjusts more quickly to  
changes in system workload for better response time results.

Other embodiments are within the scope of the following  
claims.

25           What is claimed is: